ESET

SECURE AUTHENTICATION

Guía del usuario de API

ESET **SECURE AUTHENTICATION**

Copyright ©2015 **por ESET, spol. s r.o.**
ESET Secure Authentication fue desarrollado por ESET, spol. s r.o.
Para obtener más información, visite www.eset-la.com.
Todos los derechos reservados. Ninguna parte de esta documentación podrá reproducirse, almacenarse en un sistema de recuperación o transmitirse en forma o medio alguno, ya sea electrónico, mecánico, fotocopia, grabación, escaneo o cualquier otro medio sin la previa autorización por escrito del autor.
ESET, spol. s r.o. se reserva el derecho de modificar cualquier elemento del software de la aplicación descrita sin previo aviso.

Atención al cliente: www.eset.com/support

REVISADO 9. 4. 2015

Contenido

1. Introducción 4
2. Vista general de la integración 4
3. Configuración 4
4. API de autenticación 5
4.1 Paso 1: Iniciar autenticación de 2 factores 5
4.1.1 Solicitud 5
4.1.2 Respuesta 5
4.2 Paso 2: Autenticar 6
4.2.1 Solicitud 6
4.2.2 Respuesta 6
5. API de la administración de usuarios 7
5.1 Obtener perfil de usuarios 7
5.1.1 Solicitud 7
5.1.2 Respuesta 7
5.2 Desbloquear 8
5.2.1 Solicitud 8
5.2.2 Respuesta 8
5.3 Cancelación de acceso 9
5.3.1 Solicitud 9
5.3.2 Respuesta 9
5.4 Aplicación móvil de suministro 9
5.4.1 Solicitud 9
5.4.2 Respuesta 10
5.5 Mensaje de texto de suministro 10
5.5.1 Solicitud 10
5.5.2 Respuesta 10
6. Gestión de errores 11
6.1 Errores de API 11
6.2 Errores de HTTP 11

1. Introducción

En la mayoría de las aplicaciones basadas en la web, se autentica a los usuarios antes de otorgarles acceso a los recursos protegidos. Al solicitar un factor adicional de autenticación durante el proceso de registro, dichas aplicaciones obtienen una capa adicional de seguridad.

La ESET Secure Authentication API es un servicio de internet basado en REST que puede usarse para agregar fácilmente una autenticación de dos factores (2FA) a las aplicaciones existentes.

2. Vista general de la integración

La API consta de dos terminales:

1. La Authentication API, para agregar 2FA a las aplicaciones existentes.
2. La User Management API, para administrar usuarios de 2FA.

La API funciona con métodos que son llamados por texto de publicación POSTing JSON formateado en las URLs API relevantes. Todas las respuestas son también codificadas como texto formateado con JSON que contiene el resultado del método y los mensajes de error aplicables.

La API está disponible en todos los servidores en donde el componente de Authentication Core está instalado y funciona en el protocolo seguro HTTPS en el puerto 8001. La API es un subcomponente del ESA Authentication Service. Como tal, un dominio de Active Directory y una instalación funcional de ESA son prerrequisitos para usar la API. Solo Active Directory es compatible, no se pueden usar otras tiendas de usuarios.

3. Configuración

La API está deshabilitada de manera predeterminada y debe ser habilitada antes de su uso. Cada conjunto de credenciales de API puede habilitarse para la Authentication API, la User Management API o ambas terminales. Una vez habilitada, se deben crear las credenciales API para autorizar solicitudes:

1. Inicie el ESET Secure Authentication Management Console y navegue hasta **Advanced Settings** del nodo de su dominio.
2. Expanda la sección **API**.
3. Seleccione la casilla de verificación junto a **API is enabled** y haga clic en **Save** para guardar los cambios.
4. Inicie la ESET Secure Authentication Management Console y navegue al nuevo nodo visible **API Credentials** de su dominio.
5. Haga clic en **Add Credentials** para crear un nuevo conjunto de credenciales.
6. Haga doble clic en las nuevas credenciales creadas para obtener el nombre de usuario y la contraseña que serán usados para la autenticación API. Habilite las credenciales para las terminales que se requieran:
   a. Si se debieran usar credenciales para la Authentication API, entonces marque la casilla de verificación **Enabled for Auth API**.
   b. Si se debieran usar credenciales para la User Management API, entonces marque la casilla de verificación **Enabled for User Management API**.
   c. Haga clic en **OK** para guardar los cambios.
7. Presione la **Windows tecla + R**, escriba **Services.msc** en el campo **Abierto** y luego presione **Enter** para abrir la Consola de servicios de Windows.
8. Haga clic derecho en ESET Secure Authentication **Core** y seleccione **Restart** del menú contextual.

Puede crear varios conjuntos de credenciales API. Recomendamos que cree diferentes conjuntos para cada aplicación que se proteja, como también un conjunto para pruebas.

Si la API está habilitada, todos los servidores con el componente Authentication Core instalado responderán a las solicitudes API autorizadas después de ser reiniciadas. El servicio de Authentication Core debe reiniciarse cuando se creen o eliminen las credenciales.

4. API de autenticación

Todos los métodos de Authentication API se encuentran disponibles en direcciones URL de la forma *https://127.0.0.1:8001/auth/v1/* y están protegidos ante el acceso no autorizado a través del sistema HTTP Basic Authentication, que requiere de un conjunto de API Credentials que estén habilitadas para la Authentication API antes de procesar solicitudes. El encabezado *Content-Type* debe establecerse como *application/json* para cada solicitud.

El instalador de ESET Secure Authentication usa automáticamente un certificado adecuado de seguridad SSL instalado en el equipo, o genera un certificado firmado automáticamente si no se puede encontrar otro.

El reemplazo del certificado SSL se encuentra cubierto en el documento de reemplazo del ESA API SSL Certificate.

4.1 Paso 1: Iniciar autenticación de 2 factores

Tan pronto como la aplicación existente haya verificado el nombre de usuario y la contraseña del usuario, se debe denominar el método Start 2-FA para determinar si la autenticación de dos factores ha sido habilitada para el usuario. Si se solicita, se enviará automáticamente una SMS OTP al usuario en ese momento.

4.1.1 Solicitud

Para comenzar el proceso 2FA, realice una solicitud HTTP POST en la siguiente URI:

/auth/v1/start2fa

Se debe publicar la siguiente cadena JSON:

```
{
    "username": "USERNAME"
}
```

El campo **username** es una cadena con el *samAccountName* del usuario por ser autenticado. Es muy importante que se envíe el nombre de usuario correcto a la API: el *samAccountName* es el nombre de registro normal del usuario en Active Directory.

4.1.2 Respuesta

Todas las respuestas típicas serán regresada con un código de estado(OK) HTTP 200 incluso si hubo un error en la acción de la solicitud. La respuesta será una cadena JSON. A continuación puede observar un ejemplo de una respuesta estándar:

```
{
    "expected_otp": ["APP", "SMS"],
    "error": "ERROR_NONE",
    "error_message": ""
}
```

Si no ocurrió ningún error, entonces el campo **error** mostrará “ERROR_NONE”. Consulte la sección Gestión de errores para obtener una descripción de posibles códigos de errores.

El campo **error_message** proveerá una descripción amigable del error, cuando corresponda.

El campo **expected_otp** es una matriz que especifica los tipos de OTP (contraseña de un solo uso) que pueden esperarse del usuario. Este valor puede asistir en la creación de UI, por ejemplo, indicará si el usuario debe esperar un SMS o no. Si la matriz está vacía, entonces no se requiere ninguna OTP (es decir, 2FA no está habilitado) y el usuario debe iniciar sesión inmediatamente. Los siguientes tipos de OTP pueden estar incluidos en la matriz:

- **APP** - el usuario ya ha instalado la aplicación ESA en su teléfono móvil y deberá generar una OTP con la utilización de la aplicación.
- **SMS** - el usuario no ha instalado la aplicación y se le ha enviado un SMS con una OTP.
- **HARD_TOKEN** - se le ha asignado al usuario un token de seguridad y debe generar una OTP con el dispositivo.

4.2 Paso 2: Autenticar

4.2.1 Solicitud

Para autenticar a un usuario, realice una solicitud HTTP POST en la siguiente URI:

/auth/v1/authenticate

Se debe publicar la siguiente cadena JSON:

```
{
    "username": "USERNAME",
    "otp": "123456"
}
```

El campo del **username**username es una cadena con el *samAccountName* del usuario que debe ser autenticado y el campo **otp** es una cadena con el OTP ingresado por el usuario.

4.2.2 Respuesta

Todas las respuestas típicas serán regresada con un código de estado (OK) HTTP 200 incluso si hubo un error en la acción de la solicitud. La respuesta será una cadena JSON. A continuación puede observar un ejemplo de una respuesta estándar:

```
{
    "authenticated": true,
    "error": "ERROR_NONE",
    "error_message": ""
}
```

Si no ocurrió ningún error, entonces el campo **error** mostrará **ERROR_NONE**. Consulte la sección Gestión de errores de esta guía para obtener una descripción de posibles códigos de errores.

El campo **error_message** proveerá una descripción del error si ha ocurrido un error.

El campo **authenticated** es un Boolean que especifica si la OTP proporcionada es válida. Si el valor **authenticated** es **true** la OTP del usuario ha sido validada con éxito y el usuario deberá estar registrado.

5. API de la administración de usuarios

Todos los métodos de User Management API se encuentran disponibles en direcciones URL de la forma *https://127.0.0.1:8001/manage/users/v1/* y están protegidos ante el acceso no autorizado a través del sistema HTTP Basic Authentication, que requiere de un conjunto de API Credentials que estén habilitadas para la User Management API antes de procesar solicitudes. El encabezado *Content-Type* debe establecerse como *application/json* para cada solicitud.

El instalador de ESET Secure Authentication usa automáticamente un certificado adecuado de seguridad SSL instalado en el equipo, o genera un certificado firmado automáticamente si no se puede encontrar otro.

El reemplazo del certificado SSL se encuentra cubierto en el documento de reemplazo del ESA API SSL Certificate.

5.1 Obtener perfil de usuarios

Este método devuelve la información 2FA sobre la cuenta de un usuario.

5.1.1 Solicitud

Para obtener el perfil de 2FA para un usuario, realice una solicitud HTTP GET en la siguiente URI:

/manage/users/v1/profile/USERNAME

Donde **USERNAME** es una cadena con el *samAccountName* del usuario del cuál se debe obtener el perfil. Es muy importante que se envíe el nombre de usuario correcto a la API: el *samAccountName* es el nombre de registro normal del usuario en Active Directory. El nombre de usuario debe estar cifrado en URL.

5.1.2 Respuesta

Todas las respuestas típicas serán regresada con un código de estado (OK) HTTP 200 incluso si hubo un error en la acción de la solicitud. La respuesta será una cadena JSON. A continuación puede observar un ejemplo de una respuesta estándar:


```
{
    "username": "USERNAME",
    "mobile_number": "2700000",
    "is_locked": false,
    "last_success": "2014-01-01T00:00:00",
    "last_failure": null,
    "consecutive_failures": 0,
    "credential_type": [ "APP", "SMS" ],
    "error": "ERROR_NONE",
    "error_message": ""
}
```


Si no ocurrió ningún error, entonces el campo **error** mostrará **ERROR_NONE**. Consulte la sección Gestión de errores de esta guía para obtener una descripción de posibles códigos de errores.

El campo **error_message** proveerá una descripción del error si ha ocurrido un error.

El campo **username** es una String que contiene el *samAccountName* del usuario.

El campo **mobile_number** es una String que contiene el número de móvil del usuario.

El campo **is_locked** es un Boolean que especifica si el usuario ha sido bloqueado para 2FA debido a demasiados intentos de autenticación fallidos.

El campo **last_success** es una Date que especifica la última vez que el usuario realizó una autenticación correcta. Este campo puede ser null.

El campo **last_failure** es una Date que especifica la última vez que el usuario realizó una autenticación fallida. Este campo puede ser null.

El campo **consecutive_failures** es un Integer que especifica la cantidad de intentos de autenticación fallidos consecutivos realizados por el usuario.

El campo **credential_type** es una matriz que especifica los tipos de OTP (contraseña de un solo uso) que se han habilitado para el usuario. Los siguientes tipos de OTP pueden estar incluidos en la matriz:

- **APP** - el usuario ha sido habilitado para la aplicación móvil de ESA.
- **SMS** - el usuario ha sido habilitado para SMS OTPs.
- HARD_TOKEN - el usuario ha sido habilitado para OTPs del token de seguridad.

5.2 Desbloquear

Este método desbloqueará el acceso 2FA de un usuario. No desbloqueará una cuenta bloqueada mediante Active Directory.

5.2.1 Solicitud

Para desbloquear a un usuario, realice una solicitud HTTP POST en la siguiente URI:

/manage/users/v1/unlock

Se debe publicar la siguiente cadena JSON:

```
{
    "username": "USERNAME"
}
```

El campo **username** es una cadena con el *samAccountName* del usuario por desbloquear. Es muy importante que se envíe el nombre de usuario correcto a la API: el *samAccountName* es el nombre de registro normal del usuario en Active Directory.

5.2.2 Respuesta

Todas las respuestas típicas serán regresada con un código de estado (OK) HTTP 200 incluso si hubo un error en la acción de la solicitud. La respuesta será una cadena JSON. La respuesta solo contendrá un posible código de error y mensaje, sin ningún otro dato. A continuación puede observar un ejemplo de una respuesta estándar:

```
{
    "error": "ERROR_NONE",
    "error_message": ""
}
```

Si no ocurrió ningún error, entonces el campo **error** mostrará **ERROR_NONE**. Consulte la sección Gestión de errores de esta guía para obtener una descripción de posibles códigos de errores.

El campo **error_message** proveerá una descripción del error si ha ocurrido un error.

5.3 Cancelación de acceso

Este método deshabilitará 2FA para un usuario.

5.3.1 Solicitud

Para deshabilitar 2FA para un usuario, realice una solicitud HTTP POST en la siguiente URI:

/manage/users/v1/deprovision

Se debe publicar la siguiente cadena JSON:

```
{
    "username": "USERNAME"
}
```

El campo **username** es una cadena con el *samAccountName* del usuario para deshabilitar 2FA. Es muy importante que se envíe el nombre de usuario correcto a la API: el *samAccountName* es el nombre de registro normal del usuario en Active Directory.

5.3.2 Respuesta

Todas las respuestas típicas serán regresada con un código de estado (OK) HTTP 200 incluso si hubo un error en la acción de la solicitud. La respuesta será una cadena JSON. La respuesta solo contendrá un posible código de error y mensaje, sin ningún otro dato. A continuación puede observar un ejemplo de una respuesta estándar:

```
{
    "error": "ERROR_NONE",
    "error_message": ""
}
```

Si no ocurrió ningún error, entonces el campo **error** mostrará **ERROR_NONE**. Consulte la sección Gestión de errores de esta guía para obtener una descripción de posibles códigos de errores.

El campo **error_message** proveerá una descripción del error si ha ocurrido un error.

5.4 Aplicación móvil de suministro

Este método deshabilitará las Mobile Application OTPs para el usuario. Se le enviará al usuario un mensaje de texto con la URL de instalación para la aplicación móvil.

5.4.1 Solicitud

Para proporcionar una Mobile Application para un usuario, realice una solicitud HTTP POST en la siguiente URI:

/manage/users/v1/provisionmobileapp

Se debe publicar la siguiente cadena JSON:

```
{
    "username": "USERNAME"
}
```

El campo **username** es una cadena con el *samAccountName* del usuario. Es muy importante que se envíe el nombre de usuario correcto a la API: el *samAccountName* es el nombre de registro normal del usuario en Active Directory.

5.4.2 Respuesta

Todas las respuestas típicas serán regresada con un código de estado (OK) HTTP 200 incluso si hubo un error en la acción de la solicitud. La respuesta será una cadena JSON. A continuación puede observar un ejemplo de una respuesta estándar:

```
{
        "installation_url": "http://...",
        "error": "ERROR_NONE",
        "error_message": ""
}
```

Si no ocurrió ningún error, entonces el campo **error** mostrará **ERROR_NONE**. Consulte la sección Gestión de errores de esta guía para obtener una descripción de posibles códigos de errores.

El campo **error_message** proveerá una descripción del error si ha ocurrido un error.

El campo **installation_url** es una String que contiene la URL de instalación para la Mobile Application.

5.5 Mensaje de texto de suministro

Este método habilitará a un usuario para las OTPs de mensajes de texto.

5.5.1 Solicitud

Para proporcionar OTPs de mensajes de texto para un usuario, realice una solicitud HTTP POST en la siguiente URI:

/manage/users/v1/provisiontextmessage

Se debe publicar la siguiente cadena JSON:

```
{
        "username": "USERNAME"
}
```

El campo **username** es una cadena con el *samAccountName* del usuario. Es muy importante que se envíe el nombre de usuario correcto a la API: el *samAccountName* es el nombre de registro normal del usuario en Active Directory.

5.5.2 Respuesta

Todas las respuestas típicas serán regresada con un código de estado (OK) HTTP 200 incluso si hubo un error en la acción de la solicitud. La respuesta será una cadena JSON. La respuesta solo contendrá un posible código de error y mensaje, sin ningún otro dato. A continuación puede observar un ejemplo de una respuesta estándar:

```
{
        "error": "ERROR_NONE",
        "error_message": ""
}
```

Si no ocurrió ningún error, entonces el campo **error** mostrará **ERROR_NONE**. Consulte la sección Gestión de errores de esta guía para obtener una descripción de posibles códigos de errores.

El campo **error_message** proveerá una descripción del error si ha ocurrido un error.

6. Gestión de errores

6.1 Errores de API

Todos los errores de API serán devueltos como respuesta con un código de estado HTTP 200 (OK).

El campo de **error** en la respuesta JSON indicará el código de error, el cual es un valor de cadena literal. Se definen los siguientes códigos de errores:

- **ERROR_NONE**: No ha ocurrido ningún error
- **ERROR_USER_NOT_FOUND**: El nombre de usuario ingresado no existe en el sistema
- **ERROR_FAULT**: Ha ocurrido un error no especificado

Además del campo del **error**, también se provee un **error_message** (mensaje de error) con una descripción amistosa del error. Solo se debe usar el campo del **error** para determinar las condiciones del error, ya que el campo del **error_message** es únicamente informativo y está sujeto a cambios sin aviso.

6.2 Errores de HTTP

Todos los errores de HTTP serán regresados como respuestas con un cuerpo vacío y un código de estado HTTP distinto al 200 (OK) normal.

Se pueden regresar los siguientes códigos de estado HTTP erróneos:

- **HTTP 500 (Internal Server Error)**: El servicio API encontró un error fatal desconocido
- **HTTP 400 (Bad Request)**: El formato del encabezado de “Authorization” en la solicitud HTTP no es válido
- **HTTP 401 (Unauthorized)**: No se proporcionaron credenciales API con la solicitud HTTP
- **HTTP 403 (Forbidden)**: Las credenciales presentadas con la solicitud HTTP no son válidas.